

Η άσκηση υλοποιήθηκε σε γλώσσα προγραμματισμού Java και αποτελείται από 2 διαφορετικές κλάσεις. Την κλάση Record για την αναπαράσταση ενός αντικειμένου(ID, score) και την κλάση TopK που περιέχει τον βασικό αλγόριθμο.

Η κλάση Record.java περιέχει τα πεδία id, lowerBound, totalScore, fileShown που αναπαριστούν το id του αντικειμένου, το κάτω όριο που βρήκαμε κατά την πρώτη εύρεση του σε ένα από τα σειριακά αρχεία, το τελικό του score και το σειριακό αρχείο στο οποίο βρέθηκε την πρώτη φορά. Επίσης, κάνει implement το interface Comparable και υλοποιεί την μέθοδο compareTo για να μπορεί να χρησιμοποιηθούν τα αντικείμενα της κλάσης σε ένα PriorityQueue(minheap). Στην σύγκριση δύο αντικειμένων σημαντικό ρόλο παίζει η μέθοδος **getCurrentScore()** όπου επιστρέφει το lowerBound αν δεν έχουμε βρει το totalScore του αντικειμένου, αλλιώς επιστρέφει το totalScore αν το έχουμε δει και στα δύο σειριακά αρχεία.

Στην κλάση TopK.java όπου υλοποιεί τον top-k αλγόριθμο αρχικά ο χρήστης δίνει στο command-line ως argument το k(π.χ java TopK 5). Στην συνέχεια εκτελείται η μέθοδος **readSeqFiles()**, μέσω της οποίας αφού πρώτα διαβαστούν τα δεδομένα από το rnd.txt και τοποθετηθούν στην κύρια μνήμη σε ένα array, προχωράμε στην σειριακή προσπέλαση των αρχείων seq1 και seq2. Έτσι, τοποθετούμε σε ένα hashmap τα αντικείμενα που διαβάζουμε από τα αρχεία κατά την πρώτη τους εμφάνιση και αν τα έχουμε ξαναδεί απλά ανανεώνουμε το score τους καλώντας την μέθοδο **setTotalScore()** της κλάσης Record. Αρχικοποιούμε το minheap στην **initializeMinHeap()** μόλις το μέγεθος του hashmap γίνει ίσο με το k. Μετά την αρχικοποίηση του minheap ελέγχουμε μέσω της **checkPush()** αν η τιμή της κορυφής του minheap είναι μικρότερη του αντικειμένου που διαβάζουμε την δεδομένη στιγμή και αν ναι τότε την αφαιρούμε και βάζουμε το αντικείμενο αυτό στο minheap. Επίσης, θέτουμε σε κάθε επανάληψη, μετά την προσπέλαση ενός αντικειμένου, το threshold τερματισμού T ίσο με το τελευταίο skor που έχουμε δει στο seq1.txt + τελευταίο skor που έχουμε δει στο seq2.txt + 5. Όταν το T γίνεται μικρότερο ή ίσο του κορυφαίου στοιχείου του minheap, τότε μέσω της **checkTermination()**, ελέγχουμε αν υπάρχει αντικείμενο που έχουμε δει αλλά δεν είναι στο minheap με πάνω όριο μεγαλύτερο από το κορυφαίο αντικείμενο του minheap. Αν υπάρχει τέτοιο αντικείμενο συνεχίζουμε την διαδικασία προσπελάσεων, αλλιώς ο αλγόριθμος τερματίζει. Το πάνω όριο υπολογίζεται προσθέτοντας στο κάτω όριο αυτού του αντικειμένου το τελευταίο skor που διαβάσαμε σειριακά στην πηγή όπου δεν έχουμε δει το αντικείμενο. Τέλος, όταν η διαδικασία τελειώσει καλείται η **printTopKs()** όπου αφαιρούνται επαναληπτικά τα

κορυφαία αντικείμενα από το minheap και στην συνέχεια τα τυπώνουμε στην κονσόλα σε φθίνουσα σειρά.

Εικόνα εκτέλεσης για k=5:

```
<terminated> TopK5 [Java Application] C:\Program Files\Java\
Number of sequential accesses= 3018
Top k objects:
50905: 14,84
85861: 14,76
22652: 14,74
75232: 14,74
20132: 14,74
```

Επιπλέον, για τον έλεγχο της σωστής λειτουργίας της μεθόδου δημιουργήθηκε η μέθοδος **checkBruteForce()** όπου διαβάζει σειριακά τα αντικείμενα από όλα τα αρχεία, προσθέτει τα score τους και στην συνέχεια τα τυπώνει διαταγμένα.